

### ***In the Claims***

The status of claims in the case is as follows:

1        1.    [Currently amended] A method for control and  
2        management of communication traffic, comprising the steps  
3        of:

4            expressing access rules as filters referencing system  
5            kernel data;

6            for outbound processing, determining source application  
7            indicia;

8            for inbound packet processing, executing a look-ahead  
9            function to determine target application indicia; said  
10          look-ahead function being executed within an IP layer  
11          of a protocol stack including ~~an IP layer~~ said IP  
12          layer, a transport layer, a sockets layer, and an  
13          application layer and which, for said inbound packet,  
14          said IP layer provides to said transport layer said  
15          inbound packet, marked as ~~non-deliverable~~ deny, and  
16          receives back from said transport layer indicia,

17 provided to said transport layer by said sockets layer,  
18 identifying the application layer application to which  
19 said packet would have been delivered; and

20 responsive to said source or target application  
21 indicia, executing filter processing; said filter  
22 processing including constructing and evaluating  
23 logical expressions including non-IP packet attributes  
24 of arbitrary length, and selectively using a set of  
25 logical operators, alternative filter selector fields,  
26 and value set.

1 2. [Previously presented] The method of claim 1, wherein  
2 said protocol stack is a TCP/IP protocol stack, and further  
3 comprising the steps of executing said determining and  
4 executing steps within a kernel filtering function upon  
5 encountering a filter selector field referencing kernel data  
6 not included in said packet.

1 3. [Previously presented] The method of claim 1, wherein  
2 said protocol stack is a TCP/IP protocol stack, and said  
3 filter processing including the steps of:

4 determining a task or thread identifier;

5           based on said task or thread identifier, determining a  
6           process or job identifier; and

7           based on said process or job identifier, determining  
8           job or process attributes for filter processing.

1       4.   [Previously presented] The method of claim 1, wherein  
2       said protocol stack is a TCP/IP protocol stack, and said  
3       filter processing including the steps of:

4           determining a user identifier; and

5           based on said user identifier, determining user  
6           attributes for filter processing.

1       5.   [Original] The method of claim 3, further comprising  
2       the step of determining from said task identifier a work  
3       control block containing said process or job identifier.

1       6.   [Canceled]

2       7.   [Canceled]

1 8. [Previously presented] The method of claim 1, wherein  
2 said protocol stack is a TCP/IP protocol stack, and further  
3 comprising the steps of:

4 delivering to said filters infrastructure access rules  
5 for defining security context.

1 9. [Original] The method of claim 8, said infrastructure  
2 including logging, auditing, and filter rule load controls.

1 10. [Currently amended] A method for control and  
2 management of aspects of communication traffic within  
3 filtering, comprising the steps of:

4 receiving IP packet data into a TCP/IP protocol stack  
5 executing within a system kernel;

6 for an inbound IP packet, executing a look-ahead  
7 function within an IP layer of a protocol stack  
8 including ~~an IP layer~~ said IP layer, a transport layer,  
9 a sockets layer, and an application layer and which,  
10 for said IP inbound packet, said IP layer provides to  
11 said transport layer said inbound IP packet, marked as  
12 ~~non-deliverable~~ deny, and receives back from said

13 transport layer indicia, provided to said transport  
14 layer by said sockets layer, identifying the  
15 application layer application to which said packet  
16 would have been delivered; and

17 executing filtering code within said IP layer of said  
18 system kernel with respect to non-IP packet data  
19 accessed within said system kernel outside of said  
20 TCP/IP protocol stack; said filtering code constructing  
21 and evaluating logical expressions of arbitrary length,  
22 and selectively using a set of logical operators,  
23 alternative filter selector fields, and value set.

1 11. [Original] The method of claim 10, said non-IP packet  
2 data including context data regarding said IP packet.

1 12. [Original] The method of claim 10, said non-IP packet  
2 data including data specific to a task generating said non-  
3 IP packet data.

1 13. [Original] The method of claim 10, said non-IP packet  
2 data including data specific to a task that will receive  
3 said IP packet.

1 14. [Original] The method of claim 11, said context data  
2 including packet arrival interface indicia.

15. [Canceled]

16. [Canceled]

17. [Canceled]

1 18. [Currently amended] A method for centralizing system-  
2 wide communication management and control within filter  
3 rules, comprising the steps of:

4 providing filter statements syntax for accepting  
5 parameters in the form of a selector, each selector  
6 specifying selector field, operator, and a set of  
7 values;

8 for an inbound packet, executing a look-ahead function  
9 within an IP layer of a protocol stack including ~~an IP~~  
10 said IP layer, a transport layer, a sockets layer, and  
11 an application layer and which, for said inbound  
12 packet, said IP layer provides to said transport layer  
13 said inbound packet, marked as ~~non-deliverable~~ deny,  
14 and receives back from said transport layer indicia,  
15 provided to said transport layer by said sockets layer,

16 identifying the application layer application to which  
17 said packet would have been delivered by said sockets  
18 layer;

19 said selector referencing data that does not exist in  
20 IP packets;

21 processing said filter statements, including  
22 constructing and evaluating logical expressions of  
23 arbitrary length including non-IP packet attributes,  
24 and selectively using a set of logical operators,  
25 alternative filter selector fields, and value set.

1 19. [Previously presented] The method of claim 18,  
2 wherein said protocol stack is a TCP/IP protocol stack, and  
3 said parameters selectively including userid, user profile,  
4 user class, user group, user group authority, user special  
5 authority, job name, process name, job group, job class, job  
6 priority, other job or process attributes, and date & time.

1 20. [Previously presented] The method of claim 18,  
2 wherein said protocol stack is a TCP/IP protocol stack, and  
3 said filters statements being provided within a user  
4 interface to said system.

1     21. [Previously presented] The method of claim 18,  
2     wherein said protocol stack is a TCP/IP protocol stack, and  
3     further comprising the steps of:

4             establishing a tunnel between two IP address limiting  
5             traffic to applications bound to ports at each end of  
6             said tunnel;

7             said filtering code accessing filtering attributes  
8             further limiting traffic selectively to job indicia;  
9             and

10            operating said filtering code within a kernel filtering  
11            function upon encountering a filter selector field  
12            referencing kernel data not included in said traffic.

1     22. [Currently amended] A method for traversing a portion  
2     only of a protocol stack to disallow selective IP packet  
3     traffic, comprising the steps of:

4             receiving a packet in the system kernel of the  
5             operating system of a first node from an application,  
6             said kernel including a filter processor; said filter



7 processor for constructing and evaluating logical  
8 expressions of arbitrary length including non-IP packet  
9 attributes, said logical expressions selectively  
10 including a set of logical operators, alternative  
11 filter selector fields, and value set;

12 for inbound packet processing to a first node from a  
13 second node, executing a look-ahead function in an IP  
14 layer of the system said system kernel of said first  
15 node to determine a target application; said system  
16 kernel including a TCP/IP protocol stack including ~~an~~  
17 ~~IP layer~~ said IP layer, a transport layer, a sockets  
18 layer, and an application layer and which, for said  
19 inbound packet, said IP layer provides to said  
20 transport layer said inbound packet, marked as ~~non-~~  
21 ~~deliverable~~ deny, and receives back from said transport  
22 layer indicia identifying the application layer  
23 application to which said packet would have been  
24 delivered;

25 for both said inbound packet processing, and for  
26 outbound packet processing from said first node to said  
27 second node, executing within said kernel the steps of

28           processing said packet by determining a task ID;  
  
29           responsive to said task ID, determining a  
30           corresponding work control block;  
  
31           determining a user ID, process or job identifier  
32           from said work control block;  
  
33           from the user ID, process or job identifier  
34           selectively determining attributes for said user  
35           process or job; and  
  
36           passing said attributes to said filter processor  
37           for managing and controlling communication  
38           traffic.

1       23. [Currently amended] A method for expressing access  
2       rules as filters, comprising the steps of:

3           providing a filter statements syntax for accepting  
4           parameters in the form of a selector, each selector  
5           specifying selector field, operator, and a set of  
6           values; and

7        said selector referencing data that does not exist in  
8        IP packets for controlling access to an application;

9        for an inbound IP packet, executing a look-ahead  
10       function within the IP layer of a protocol stack  
11       including ~~[[an]]~~ said IP layer, a transport layer, a  
12       sockets layer, and an application layer and which, for  
13       said IP inbound packet, said IP layer provides to said  
14       transport layer said inbound IP packet, marked as ~~non-~~  
15       ~~deliverable~~ deny, and receives back from said transport  
16       layer indicia, provided to said transport layer by said  
17       sockets layer, identifying the application layer  
18       application to which said packet would have been  
19       delivered; and

20       processing said filter statements by constructing and  
21       evaluating logical expressions including non-IP packet  
22       attributes of arbitrary length, said logical  
23       expressions selectively including a set of logical  
24       operators, alternative filter selector fields, and  
25       value set referencing said application layer  
26       application.

1       24. [Currently amended] A method for managing and

2 controlling communication traffic by centralizing access  
3 rules in filters including non-IP packet attributes  
4 executing within and referencing data available in system  
5 kernels, comprising the steps for outbound packet processing  
6 from a first node to a second node of:

7 receiving said packet in the kernel of the operating  
8 system of said first node from an application or  
9 process at said first node;

10 processing said packet by determining a task ID;

11 responsive to said task ID, determining a corresponding  
12 work control block;

13 responsive to said work control block, determining a  
14 process or job identifier;

15 responsive to said process or job identifier,  
16 determining job or process attributes; and

17 executing said filters by constructing and evaluating  
18 logical expressions of arbitrary length, said logical  
19 expressions selectively including a set of logical

20 operators, alternative filter selector fields, and  
21 value set.

1 25. [Currently amended] The method of claim 24, further  
2 comprising the steps for inbound packet processing from said  
3 second node to said first node of:

4 initially operating said kernel at said first node to  
5 determine a target application for said packet at said  
6 first node by executing a look-ahead function within  
7 the IP layer of a protocol stack including [[an]] said  
8 IP layer, a transport layer, a sockets layer, and an  
9 application layer and which, for said inbound packet,  
10 said IP layer provides to said transport layer said  
11 inbound packet, marked as ~~non-deliverable~~ deny, and  
12 receives back from said transport layer indicia,  
13 provided to said transport layer by said sockets layer,  
14 identifying the application layer application to which  
15 said packet would have been delivered;.

26. [Canceled]

27. [Canceled]

28. [Canceled]

1       29. [Currently amended] A method for managing and  
2       controlling communication traffic by centralizing the access  
3       rules, comprising the steps for outbound packet processing  
4       from a first node to a second node of:

5           receiving said packet in the kernel of the operating  
6           system of said first node from an application or  
7           process at said first node, said kernel including a  
8           filter processor for constructing and evaluating  
9           logical expressions including non-IP packet attributes  
10          of arbitrary length, said logical expressions  
11          selectively including a set of logical operators,  
12          alternative filter selector fields, and value set;

13          processing said packet within the IP layer of a TCP/IP  
14          stack;

15                by determining a task ID;

16                responsive to said task ID, determining a  
17                corresponding work control block;

18                determining a user ID control block from said work  
19                control block;

20 from the user ID control block determining  
21 attributes for said user; and  
  
22 passing said attributes to said filter processor  
23 for managing and controlling communication  
24 traffic.

1 30. [Currently amended] The method of claim 29, further  
2 comprising the steps for inbound packet processing from said  
3 second node to said first node of:

4 initially operating said kernel at said first node to  
5 determine a target application for said packet at said  
6 first node by executing a look-ahead function within  
7 said IP layer of said TCP/IP protocol stack, said  
8 TCP/IP protocol stack including [[an]] said IP layer, a  
9 transport layer, a sockets layer, and an application  
10 layer and which, for said inbound packet, said IP layer  
11 provides to said transport layer said inbound packet,  
12 marked as ~~non-deliverable~~ deny, and receives back from  
13 said transport layer indicia, provided to said  
14 transport layer by said sockets layer, identifying the  
15 application layer application to which said packet  
16 would have been delivered.

31. [Canceled]

32. [Canceled]

33. [Canceled]

1 34. [Currently amended] A method for control and  
2 management of communication traffic with respect to a system  
3 node, comprising the steps of:

4 receiving at said system node an inbound packet; and

5 executing within a protocol stack of the system kernel  
6 of said system node a filtering function identifying  
7 for said inbound packet a filter including non-IP  
8 packet attributes referencing non-packet data, and  
9 constructing and evaluating logical expressions of  
10 arbitrary length, said logical expressions selectively  
11 including a set of logical operators, alternative  
12 filter selector fields, and value set; and

13 responsive to said filter, executing a look-ahead  
14 function for identifying a target application for said  
15 inbound packet; said look-ahead function executed  
16 within the IP layer of a protocol stack including



17        [[an]] said IP layer, a transport layer, a sockets  
18        layer, and an application layer and which, for said IP  
19        inbound packet, said IP layer provides to said  
20        transport layer said inbound packet, marked as ~~non-~~  
21        ~~deliverable~~ deny, and receives back from said transport  
22        layer indicia, provided to said transport layer by said  
23        sockets layer, identifying the application layer  
24        application to which said packet would have been  
25        delivered[[:]].

1        35. [Currently amended] The look-ahead function of the  
2        method of claim 34 wherein said protocol stack is a TCP/IP  
3        protocol stack, and further comprising the steps of:

4        passing to a transport layer function identified by an  
5        IP header a packet marked ~~non-deliverable~~ deny for  
6        determining which user-level process or job is to  
7        receive said packet;

8        receiving from said transport layer an application  
9        layer task identifier for said user-level process or  
10        job; and thereafter

11        passing said packet marked by said task identifier to

12           said transport layer for delivery to said application  
13           layer task.

1       36. [Currently amended] System for control and management  
2       of communication traffic, comprising:

3           a system kernel including a filter function and stack  
4           data;

5           said filter function including a filter including non-  
6           IP packet attributes selectively referencing said stack  
7           data for expressing access rules;

8           said filter function being responsive to receipt of an  
9           outbound packet for determining a source application;

10          said filter function being responsive to receipt of an  
11          inbound packet ~~processing~~ for executing a look-ahead  
12          function within the IP layer of a TCP/IP protocol stack  
13          to determine a target application; said protocol stack  
14          including [[an]] said IP layer, a transport layer, a  
15          sockets layer, and an application layer and which, for  
16          said inbound packet, said IP layer provides to said  
17          transport layer said inbound packet, marked as ~~non-~~

18       ~~deliverable~~ deny, and receives back from said transport  
19       layer indicia, provided to said transport layer by said  
20       sockets layer, identifying the application layer  
21       application to which said packet would have been  
22       delivered; and

23       said filter function being responsive to said source or  
24       target application for executing filter processing  
25       including constructing and evaluating logical  
26       expressions of arbitrary length, said logical  
27       expressions selectively including a set of logical  
28       operators, alternative filter selector fields, and  
29       value set.

1       37. [Currently amended] A system for control and  
2       management of aspects of communication traffic within  
3       filtering, comprising:

4       a system kernel;

5       a protocol stack including an IP layer, a transport  
6       layer, a sockets layer, and an application layer for  
7       executing within said IP layer of said system kernel,  
8       responsive to an inbound IP packet, a look-ahead

9 function by which said IP layer provides to said  
10 transport layer said inbound IP packet, marked as ~~non-~~  
11 ~~deliverable~~ deny, and receives back from said transport  
12 layer indicia, provided to said transport layer by said  
13 sockets layer, identifying the application layer  
14 application to which said packet would have been  
15 delivered; and

16 filtering code within said system kernel operable with  
17 respect to non-IP packet data accessed within said  
18 system kernel outside of said protocol stack for  
19 controlling and managing said aspects of communication  
20 traffic; said filter code for constructing and  
21 evaluating logical expressions of arbitrary length  
22 including non-IP packet attributes, said logical  
23 expressions selectively including a set of logical  
24 operators, alternative filter selector fields, and  
25 value set.

1 38. [Currently amended] A system for centralizing system-  
2 wide communication management and control within filter  
3 rules including non-IP packet attributes, comprising:

4 filter statements having a syntax for accepting

parameters in the form of a selector, each selector specifying selector field, operator, and a set of values;

said selector referencing data that does not exist in IP packets;

a look-ahead function within the IP layer of a protocol stack including ~~[[an]]~~ said IP layer, a transport layer, a sockets layer, and an application layer which, for an inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, for identifying the application layer application to which said packet would have been delivered; and

a filter processor for constructing and evaluating filter statements including logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set.

1     39. [Currently amended] A system for traversing a portion  
2     only of a TCP/IP protocol stack to disallow selective IP  
3     packet traffic, comprising:

4         a system kernel;

5         a filter processor executing within said system kernel  
6         for constructing and evaluating logical expressions of  
7         arbitrary length, said logical expressions selectively  
8         including a set of logical operators, alternative  
9         filter selector fields including non-IP packet  
10        attributes, and value set;

11        said filter processor responsive to an inbound packet  
12        for executing within an IP layer a look-ahead function  
13        for determining a target application; said look-ahead  
14        function operating within said IP layer of said TCP/IP  
15        protocol stack including [[an]] said IP layer, a  
16        transport layer, a sockets layer, and an application  
17        layer and which, for said IP inbound packet, said IP  
18        layer provides to said transport layer said inbound IP  
19        packet, marked as ~~non-deliverable~~ deny, and receives  
20        back from said transport layer indicia, provided to  
21        said transport layer by said sockets layer, identifying

22 the application layer application to which said packet  
23 would have been delivered;

24 said filter processor responsive to both inbound and  
25 outbound packets for

26 processing said packet by determining a task ID;

27 responsive to said task ID, determining a  
28 corresponding work control block;

29 determining a user ID, process or job identifier  
30 from said work control block;

31 from the user ID, process or job identifier  
32 selectively determining attributes for said user  
33 process or job; and

34 passing said attributes to said filter processor  
35 for managing and controlling communication  
36 traffic.

1 40. [Currently amended] A system for expressing access  
2 rules as filters, comprising:

3 filter statements for accepting parameters in the form  
4 of a selector, each selector specifying selector field,  
5 operator, and a set of values;

6 said selector referencing data that does not exist in  
7 IP packets for controlling access to an application;

8 a look-ahead function executing within the IP layer of  
9 a protocol stack including [[an]] said IP layer, a  
10 transport layer, a sockets layer, and an application  
11 layer and which, for an inbound packet, said IP layer  
12 provides to said transport layer said inbound packet,  
13 marked as ~~non-deliverable~~ deny, and receives back from  
14 said transport layer indicia, provided to said  
15 transport layer by said sockets layer, identifying the  
16 application layer application to which said packet  
17 would have been delivered; and

18 a filter processor for constructing and evaluating said  
19 filter statements as logical expressions of arbitrary  
20 length, each said logical expression selectively  
21 including said operator selected from a set of logical  
22 operators, alternative filter selector fields including  
23 non-IP packet attributes, and value set.



1     41. [Currently amended] A system for managing and  
2     controlling communication traffic by centralizing access  
3     rules in filters including non-IP packet attributes  
4     executing within and referencing data available in system  
5     kernels, comprising:

6             a computer readable medium;

7             first code for receiving a packet in the kernel of the  
8             operating system of a first node from an application or  
9             process at said first node; said kernel responsive to  
10            an inbound packet, for executing a look-ahead function  
11            within the IP layer of a TCP/IP protocol stack  
12            including [[an]] said IP layer, a transport layer, a  
13            sockets layer, and an application layer and which, for  
14            said inbound packet, said IP layer provides to said  
15            transport layer said inbound IP packet, marked as ~~non-~~  
16            ~~deliverable~~ deny, and receives back from said transport  
17            layer indicia, provided to said transport layer by said  
18            sockets layer, identifying the application layer  
19            application to which said packet would have been  
20            delivered;

21           second code for processing said packet by determining a  
22           task ID;  
  
23           third code responsive to said task ID for determining a  
24           corresponding work control block;  
  
25           fourth code responsive to said work control block for  
26           determining a process or job identifier;  
  
27           fifth code responsive to said process or job identifier  
28           for determining job or process attributes;  
  
29           sixth code for executing said filters by constructing  
30           and evaluating logical expressions of arbitrary length,  
31           said logical expressions selectively including a set of  
32           logical operators, alternative filter selector fields,  
33           and value set; and wherein  
  
34           said first, second, third, fourth, fifth, and sixth  
35           code is recorded on said computer readable medium.

42.   [Canceled]

1     43.   [Currently amended]   A system for control and

management of communication traffic with respect to a system node, comprising:

a filtering function executing within the IP layer of a protocol stack of the system kernel of said system node identifying for an inbound packet a filter referencing non-packet data; and

a look-ahead function responsive to said filter including non-IP packet attributes for identifying a target application for said inbound packet; said look-ahead function functioning within said IP layer of [[a]] said protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;; and

a filter processor for constructing and evaluating logical expressions of arbitrary length, said logical

23 expressions selectively including a set of logical  
24 operators, alternative filter selector fields, and  
25 value set.

44. [Canceled]

1 45. [Currently amended] A computer program product for  
2 control and management of aspects of communication traffic  
3 within filtering, said computer program product comprising:

4 a computer readable medium;

5 first program instructions to receive IP packet data  
6 into a TCP/IP protocol stack executing within a system  
7 kernel including, for processing an inbound IP packet,  
8 a look-ahead function within the IP layer of a protocol  
9 stack including ~~[[an]]~~ said IP layer, a transport  
10 layer, a sockets layer, and an application layer and  
11 which, for said IP inbound packet, said IP layer  
12 provides to said transport layer said inbound IP  
13 packet, marked as ~~non-deliverable~~ deny, and receives  
14 back from said transport layer indicia, provided to  
15 said transport layer by said sockets layer, identifying  
16 the application layer application to which said packet

17           would have been delivered;

18           second program instructions to execute filtering code  
19           within said system kernel with respect to non-IP packet  
20           data accessed within said system kernel outside of said  
21           TCP/IP protocol stack by constructing and evaluating  
22           logical expressions of arbitrary length, said logical  
23           expressions selectively including a set of logical  
24           operators, alternative filter selector fields, and  
25           value set; and wherein

26           said first and second program instructions are recorded  
27           on said medium.

1       46. [Currently amended] A computer program product for  
2       centralizing system-wide communication management and  
3       control within filter rules, said computer program product  
4       comprising:

5           a computer readable medium;

6           first program instructions to execute filter statements  
7           including non-IP packet attributes having a syntax for  
8           accepting parameters in the form of a selector, each

9 selector specifying selector field, a logical operator  
10 selected from a set of a plurality of logical  
11 operators, and a set of values; and

12 second program instructions to cause said selector to  
13 reference data that does not exist in IP packets, said  
14 data including application layer indicia obtained for  
15 an incoming packet by a look-ahead function; said look-  
16 ahead function executing within the IP layer of a  
17 protocol stack including [[an]] said IP layer, a  
18 transport layer, a sockets layer, and an application  
19 layer and which, for said IP inbound packet, said IP  
20 layer provides to said transport layer said inbound IP  
21 packet, marked as ~~non-deliverable~~ deny, and receives  
22 back from said transport layer indicia, provided to  
23 said transport layer by said sockets layer, identifying  
24 the application layer application to which said packet  
25 would have been delivered; and wherein

26 said first and second program instructions are recorded  
27 on said medium.

1 47. [Currently amended] A computer program product for  
2 managing and controlling communication traffic by

centralizing access rules in filters including non-IP packet attributes executing within and referencing data available in system kernels, said computer program product comprising:

a computer readable medium;

first program instructions to receive said packet in the kernel of the operating system of said first node from a process at said first node;

second program instructions to process said packet by determining a task ID;

third program instructions, responsive to said task ID, to determine a corresponding work control block;

fourth program instructions, responsive to said work control block, to determine a process or job identifier;

fifth program instructions, responsive to said process or job identifier, to determine job or process attributes; and

20 sixth program instructions to execute a filter  
21 processor for constructing and evaluating logical  
22 expressions of arbitrary length, said logical  
23 expressions selectively including a set of logical  
24 operators, alternative filter selector fields including  
25 non-IP packet attributes, and value set; and wherein  
  
26 said first, second, third, fourth, fifth, and sixth  
27 program instructions are recorded on said medium.

1 48. [Currently amended] The computer program product of  
2 claim 47, wherein said protocol stack is a TCP/IP protocol  
3 stack, and said computer program product further comprising  
4 for inbound packet processing from said second node to said  
5 first node:

6 sixth program instructions to initially operate said  
7 kernel at said first node to determine a target  
8 application for said packet at said first node by  
9 executing a look-ahead function within the IP layer of  
10 a protocol stack including [[an]] said IP layer, a  
11 transport layer, a sockets layer, and an application  
12 layer and which, for said IP inbound packet, said IP  
13 layer provides to said transport layer said inbound IP



14 packet, marked as ~~non-deliverable~~ deny, and receives  
15 back from said transport layer indicia, provided to  
16 said transport layer by said sockets layer, identifying  
17 the application layer application to which said packet  
18 would have been delivered;; and wherein

19 said sixth program instructions are recorded on said  
20 medium.

1 49. [Currently amended] A computer program product for  
2 control and management of communication traffic, comprising:

3 a computer readable medium;

4 first program instructions for expressing access rules  
5 as filters including non-IP packet attributes  
6 referencing system kernel data;

7 second program instructions, for outbound processing,  
8 for determining a source application;

9 third program instructions, for inbound packet  
10 processing, for executing a look-ahead function to  
11 determine a target application; said look-ahead

function operating within the IP layer of a protocol stack including ~~[[an]]~~ said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;

fourth program instructions, selectively responsive to said source and target application, for executing filter processing including constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set; and wherein

said first, second, third, and fourth program instructions are recorded on said computer readable medium.

50. [Currently amended] A computer program product for

2 control and management of aspects of communication traffic  
3 within filtering, comprising:

4 a computer readable medium;

5 first program instructions for receiving IP packet data  
6 into a TCP/IP protocol stack including an IP layer  
7 executing within a system kernel;

8 second program instructions for executing filtering  
9 code within said IP layer of said system kernel with  
10 respect to non-IP packet data accessed within said  
11 system kernel outside of said TCP/IP protocol stack;  
12 said filtering code constructing and evaluating logical  
13 expressions of arbitrary length, said logical  
14 expressions selectively including a set of logical  
15 operators, alternative filter selector fields including  
16 non-IP packet attributes, and value set; and wherein

17 said first and second program instructions are recorded  
18 on said computer readable medium.

1 51. [Currently amended] A computer program element for  
2 centralizing system-wide communication management and

control within filter rules, comprising:

a computer readable medium;

first program instructions for providing filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, a logical operator, and a set of values,

second program instructions for executing filtering by constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including said logical operator selected from a set of logical operators, at least one said selector field including non-IP packet attributes, and at least one said value;

said selector referencing data that does not exist in IP packets including data obtained, for an inbound IP packet, by executing a look-ahead function within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer

23           said inbound IP packet, marked as ~~non-deliverable~~ deny,  
24           and receives back from said transport layer indicia,  
25           provided to said transport layer by said sockets layer,  
26           identifying the application layer application to which  
27           said packet would have been delivered; and wherein

28           said first and second program instructions are recorded  
29           on said computer readable medium.

1       52. [Currently amended] A computer program product for  
2       managing and controlling communication traffic by  
3       centralizing access rules in filters on non-IP packet  
4       attributes executing within, and referencing data available  
5       in, system kernels, comprising:

6           a computer readable medium;

7           first program instructions for receiving said packet in  
8           the kernel of the operating system of said first node  
9           from an application or process at said first node;

10          second program instructions for processing said packet  
11          by determining a task ID;

12           third program instructions, responsive to said task ID,  
13           for determining a corresponding work control block;

14           fourth program instructions, responsive to said work  
15           control block, for determining a process or job  
16           identifier;

17           fifth program instructions, responsive to said process  
18           or job identifier, for determining job or process  
19           attributes;

20           sixth program instructions for executing a filter  
21           processor for constructing and evaluating logical  
22           expressions of arbitrary length, said logical  
23           expressions selectively including a set of logical  
24           operators, alternative filter selector fields, and  
25           value set; and wherein

26           said first, second, third, fourth, fifth, and sixth  
27           program instructions are recorded on said computer  
28           readable medium.

1       53. [Currently amended] The computer program product of  
2       claim 52, further comprising for inbound packet processing

3 from said second node to said first node:

4 seventh program instructions initially operating said  
5 kernel at said first node to determine a target  
6 application for said packet at said first node by  
7 executing a look-ahead function within the IP layer of  
8 a protocol stack including [[an]] said IP layer, a  
9 transport layer, a sockets layer, and an application  
10 layer and which, for said IP inbound packet, said IP  
11 layer provides to said transport layer said inbound IP  
12 packet, marked as ~~non-deliverable~~ deny, and receives  
13 back from said transport layer indicia, provided to  
14 said transport layer by said sockets layer, identifying  
15 the application layer application to which said packet  
16 would have been delivered; and wherein

17 said seventh program instructions are recorded on said  
18 computer readable medium.